

REMARKS

The Examiner's Action mailed on August 15, 2008, has been received and its contents carefully considered.

Claims 1 and 9 are the independent claims, and claims 1-14 remain pending in the application. For at least the following reasons, it is submitted that this application is in condition for allowance.

Claims 1-14 were rejected under 35 U.S.C. §103(a) as being obvious over *Alexander* (U.S. 6,553,029 B1) in view of *Tursich* (U.S. 6,671,828 B1). This rejection is respectfully traversed.

Claim 1 recites:

A multiple port single chip Ethernet switch comprising at least the following component parts:

- a physical layer entity (PHY) including a plurality of ports;
- an address table for being written to and read out information to operate the plurality of ports;
- a switch for switching the Ethernet switch to a daisy chain test mode; and
- an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip;

 wherein said component parts of said Ethernet switch are formed on said single chip.

Claim 9 recites:

A daisy chain test for a single chip Ethernet switch integrated with a physical layer entity including a plurality of ports, the switch having an address table for being written to and read out information to operate the plurality of ports, the test comprising the steps of:

connecting each of the plurality of ports to a respective passive loop-back device;

selecting a start transmission port and a stop receiving port from the plurality of ports;

supplying a test packet to the start transmission port; and

proceeding a packet source address learning process for delivering the test packet from the start transmission port to the stop receiving port progressively, wherein the step of proceeding employs a source address learning engine with a daisy chain testing function; and

determining a test result by verifying a last received packet at the stop receiving port.

Alexander discloses control operations required to transfer packets to and from physical links in an Ethernet switch. See col. 4, lines 13-16:

The discussion will focus on the control operations required to transfer packets to and from these physical links, when some or all of these physical links are being aggregated into larger logical links.

However, *Alexander* fails to disclose a test for an Ethernet switch. In *Alexander*, a link aggregation mechanism comprises address resolution unit **10**, MAC address look-up table **12**, embedded CPU **14** and queuing unit **18**. See FIG. 1 and col. 3, lines 44-53:

FIG. 1 is a block diagram of a hardware and firmware link aggregation mechanism in accordance with the invention.

FIG. 2 is flowchart which illustrates the general packet forwarding flow methodology of the invention.

DESCRIPTION

As shown in FIG. 1, the hardware elements constituting the link aggregation packet datapath comprise address resolution unit **10**, MAC address look-up table **12**, embedded CPU **14** and queuing unit **18**.

The address resolution unit **10** accepts Ethernet packets from multiple incoming packet streams, and the queuing unit **18** outputs the Ethernet packets from multiple outgoing packet streams. See col. 4, lines 17-51:

Address resolution unit **10** accepts multiple streams of packets (one stream from each physical link: FIG. 2, block **24**) and performs an address look-up process on each packet using MAC address look-up table **12** (FIG. 2, block **26**). Address lookup table **12** contains a set of MAC addresses known to the Ethernet switch as well as context information that must be associated with the MAC addresses for use in packet forwarding. (The contents of address look-up table **12** are generated by firmware running on embedded CPU **14**, as hereinafter explained.) Address resolution unit **10** thus extracts the source and destination MAC addresses from each Ethernet packet, looks up the addresses in MAC address look-up table **12** to determine if associated contexts exist for the source and destination MAC addresses, and also retrieves the contexts from table **12** if they exist. Address resolution unit **10** then presents the packet data along with the source and destination MAC address contexts (if found) to embedded CPU **14** for processing by packet forwarding firmware routine **20** (FIG. 2, block **28**).

Embedded CPU **14** implements packet forwarding firmware routine **20** to perform the actual control decisions required to forward the packet. Two different firmware routines **16**, **20** are executed. Address table creation ("learning function") firmware routine **16** is invoked when address resolution unit **10** determines that a particular source MAC address found within a packet is not present in address look-up table **12**; in which case table **12** is updated accordingly. Packet forwarding ("forwarding function") firmware routine **20** is executed for every packet, to produce the actual forwarding command (i.e., the decision as to which physical port or ports the packet must be forwarded to) supplied to queuing unit **18**. Packet forwarding routine **20** utilizes distribution table **22** to select among ports that have been grouped into aggregates.

Specifically, a packet is received from the address resolution unit 10 and outputted from the queuing unit 18, the packet is never delivered through the multiple incoming packet streams and the multiple outgoing packet streams progressively from a start transmission port to a stop receiving port to test the link aggregation mechanism. Further, the learning function of *Alexander* is used to update the address look-up table 12 when the address resolution unit 10 determines that a particular source MAC address found within a packet is not present in the address look-up table 12. See col. 4, lines 37-45 (also included in text quoted above):

Embedded CPU 14 implements packet forwarding firmware routine 20 to perform the actual control decisions required to forward the packet. Two different firmware routines 16, 20 are executed. Address table creation ("learning function") firmware routine 16 is invoked when address resolution unit 10 determines that a particular source MAC address found within a packet is not present in address look-up table 12; in which case table 12 is updated accordingly.

Alexander fails to disclose that the learning function performs a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip. In other words, although *Alexander* mentions an "address resolution unit", *Alexander* fails to disclose "a source address learning engine for performing a packet source address learning process under the daisy chain test mode" as recited in Claim 1 or "a source address learning engine with a daisy chain testing function" as recited in Claim 9.

Hence, nowhere does *Alexander* disclose "a switch for switching the Ethernet switch to a daisy chain test mode" or "an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip" as recited in Claim 1.

Similarly, nor does *Alexander* disclose "proceeding a packet source address learning process for delivering the test packet from the start transmission port to the stop receiving port progressively, wherein the step of proceeding employs a source address learning engine with a daisy chain testing function" or "determining a test result by verifying a last received packet at the stop receiving port" as recited in Claim 9.

The Office Action admits that *Alexander* fails to disclose a daisy chain test mode, and relies upon *Tursich* to supply this deficiency.

In *Tursich*, the test system screens packets entering the daisy-chain based on the destination addresses in the packets to control the traffic in a daisy-chain of protocol analyzers, and therefore a cost-effective test system is created. See col. 1, line 65 to col. 2, line 5:

The invention solves the above problem with a cost-effective test system that controls the traffic in a daisy-chain of protocol analyzers. The test system screens packets entering the daisy-chain based on the destination addresses in the packets. Advantageously, the system is implemented with simple control processing that does not add significant cost or complexity to the packet network or to the protocol analyzers.

More specifically, *Tursich only discloses that protocol analyzers 110, 120 and 130 are daisy-chained together*. See FIG. 1 and col. 2, lines 53-54:

Protocol analyzers 110, 120, and 130 are daisy-chained together.

Tursich fails to disclose a daisy chain test mode. Moreover, the test packets of *Tursich* are generated by the protocol analyzers 110, 120, and 130 and transferred to the test computer 140. The test computer 140 processes the test packets to generate test results regarding target communication device 150 for network operators, and also transfers test packets to protocol analyzers 110, 120, and 130 to control testing. The protocol analyzer 130 also receives packets carried by packet network 152, and processes the packets internally based on the destination addresses in the packets, transfers some packets to protocol analyzers 110 and 120, and deletes the other packets. See ABSTRACT and col. 2, line 59 to col. 3, line 3:

ABSTRACT

A protocol analyzer for a test system has a control system coupled to a first interface and a second interface. The first interface receives a packet from a packet network. The control system receives the packet from the first interface and either deletes the packet or transfers the packet to the second interface based on a destination address in the packet. The second interface transfers the packet to another protocol analyzer. The destination address could be a MAC address.

Protocol analyzers 110, 120, and 130 monitor traffic on lines-under-test 151 to generate and transfer test packets to test computer 140. Test computer 140

processes the test packets to generate test results regarding target communication device 150 for network operators. Test computer 140 also transfers test packets to protocol analyzers 110, 120, and 130 to control testing. Protocol analyzer 130 receives the test packets from test computer 140. Protocol analyzer 130 also receives other packets carried by packet network 152. Protocol analyzer 130 processes some packets internally, transfers some packets to protocol analyzers 110 and 120, and deletes the other packets.

Tursich therefore fails to disclose a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip.

Hence, *Tursich* also fails to disclose “a switch for switching the Ethernet switch to a daisy chain test mode” or “an address resolution control logic including a source address learning engine for performing a packet source address learning process under the daisy chain test mode to deliver a test packet through the plurality of ports progressively from a start transmission port to a stop receiving port to test the chip” as recited in Claim 1.

Similarly, nor does *Tursich* disclose “proceeding a packet source address learning process for delivering the test packet from the start transmission port to the stop receiving port progressively, wherein the step of proceeding employs a source address learning engine with a daisy chain testing function” or “determining a test result by verifying a last received packet at the stop receiving port” as recited in Claim 9.

Consequently, neither *Alexander* nor *Tursich*, whether taken separately or in combination, teaches or suggests all the features recited in claim 1 or claim 9,

and therefore claims 1 and 9 are allowable, together with claims 2-8 and 10-14 that depend respectively therefrom.

It is submitted that this application is in condition for allowance. Such action and the passing of this case to issue are requested.

Should the Examiner feel that a conference would help to expedite the prosecution of this application, the Examiner is hereby invited to contact the undersigned counsel to arrange for such an interview.

Should the remittance be accidentally missing or insufficient, the Commissioner is hereby authorized to charge the fee to our Deposit Account No. 18-0002, and advise us accordingly.

Respectfully submitted,



Alun L. Palmer – Reg. No. 47,838
RABIN & BERDO, PC – Cust. No. 23995
Facsimile: 202-408-0924
Telephone: 202-371-8976

October 28, 2008
Date

ALP/pq

RESPONSE

10/619,144